Creation of binary tree using linked list

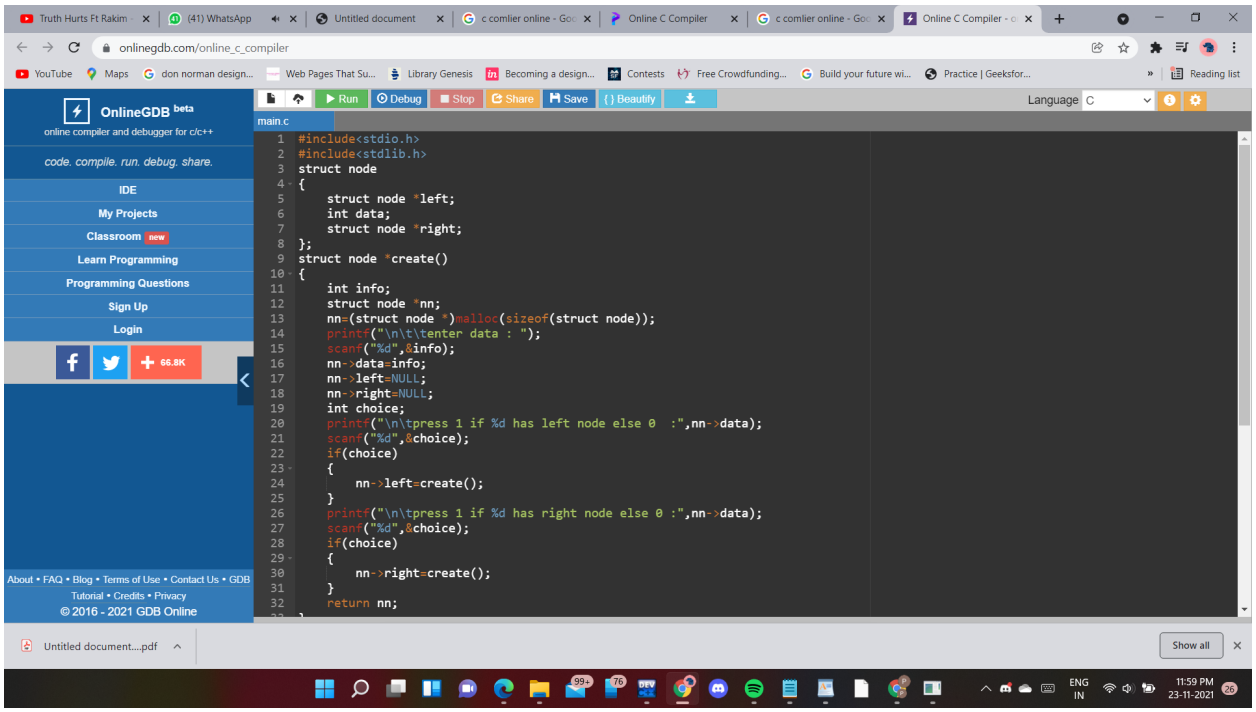
Submitted by: PRANJALI SINGH

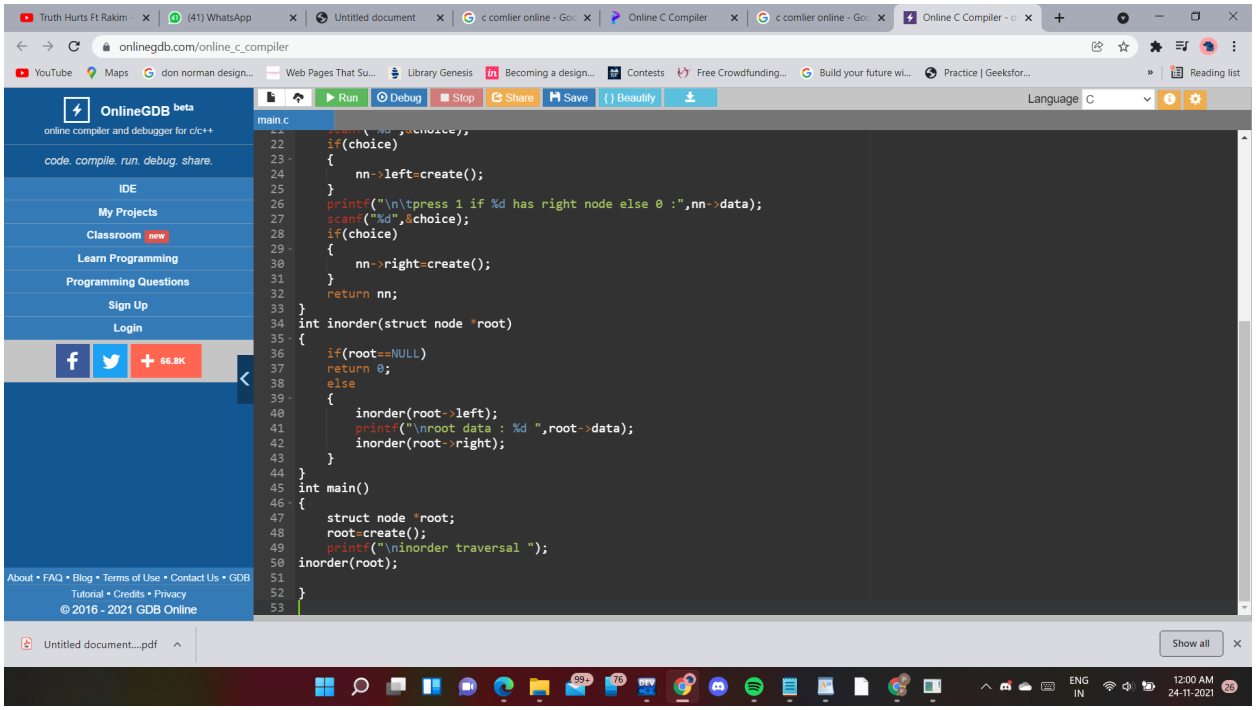
SOLUTION-1

INPUT:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
	struct node *left;
	int data;
	struct node *right;
};
struct node *create()
{
	int info;
	struct node *nn;
	nn=(struct node *)malloc(sizeof(struct node));
	printf("\n\t\tenter data : ");
	scanf("%d",&info);
	nn->data=info;
	nn->left=NULL;
	nn->right=NULL;
	int choice;
	printf("\n\tpress 1 if %d has left node else 0  :",nn->data);
	scanf("%d",&choice);
	if(choice)
	{
		nn->left=create();
	}
	printf("\n\tpress 1 if %d has right node else 0 :",nn->data);
	scanf("%d",&choice);
	if(choice)
	{
		nn->right=create();
	}
	return nn;
}
int inorder(struct node *root)
{
	if(root==NULL)
	return 0;
```

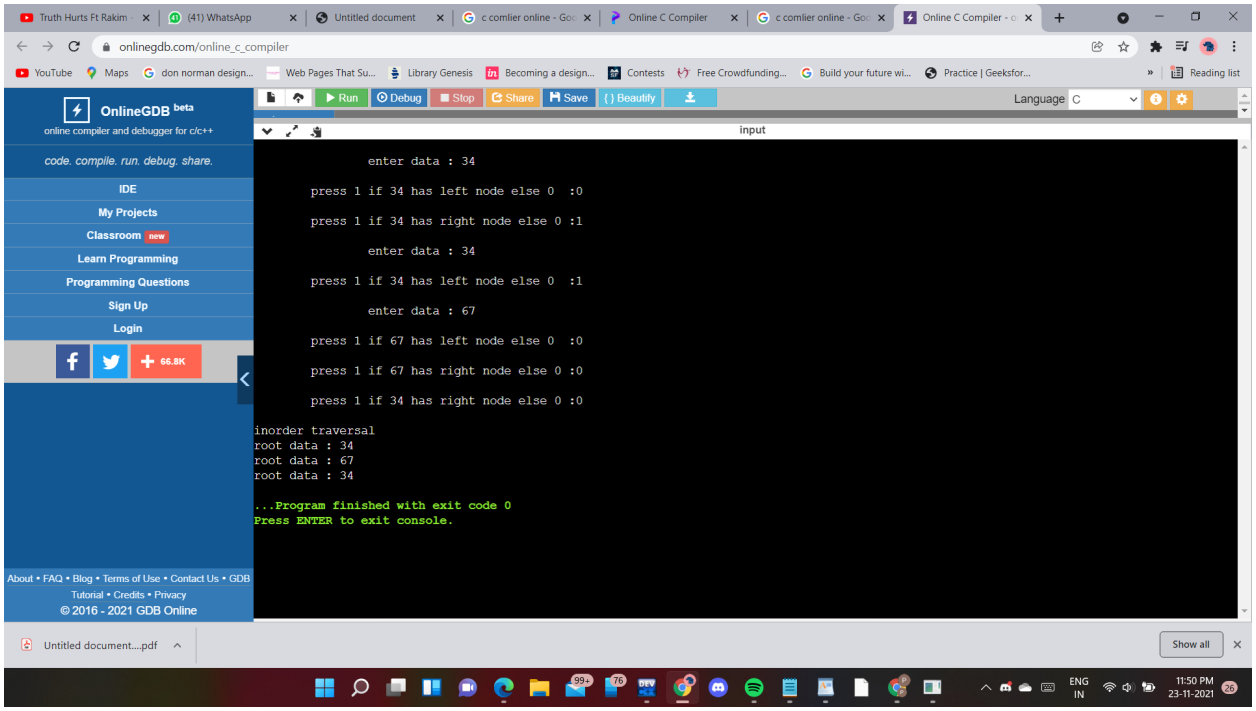
```
        else
        {
                inorder(root->left);
                printf("\nroot data : %d ",root->data);
                inorder(root->right);
        }
}
int main()
{
        struct node *root;
        root=create();
        printf("\ninorder traversal ");
inorder(root);

}
```





```
    if(choice)
    {
        nn->left=create();
    }
    printf("\n\tpress 1 if %d has right node else 0 :",nn->data);
    scanf("%d",&choice);
    if(choice)
    {
        nn->right=create();
    }
    return nn;
}
int inorder(struct node *root)
{
    if(root==NULL)
    return 0;
    else
    {
        inorder(root->left);
        printf("\nroot data : %d ",root->data);
        inorder(root->right);
    }
}
int main()
{
    struct node *root;
    root=create();
    printf("\ninorder traversal ");
inorder(root);

}
```

OUTPUT:



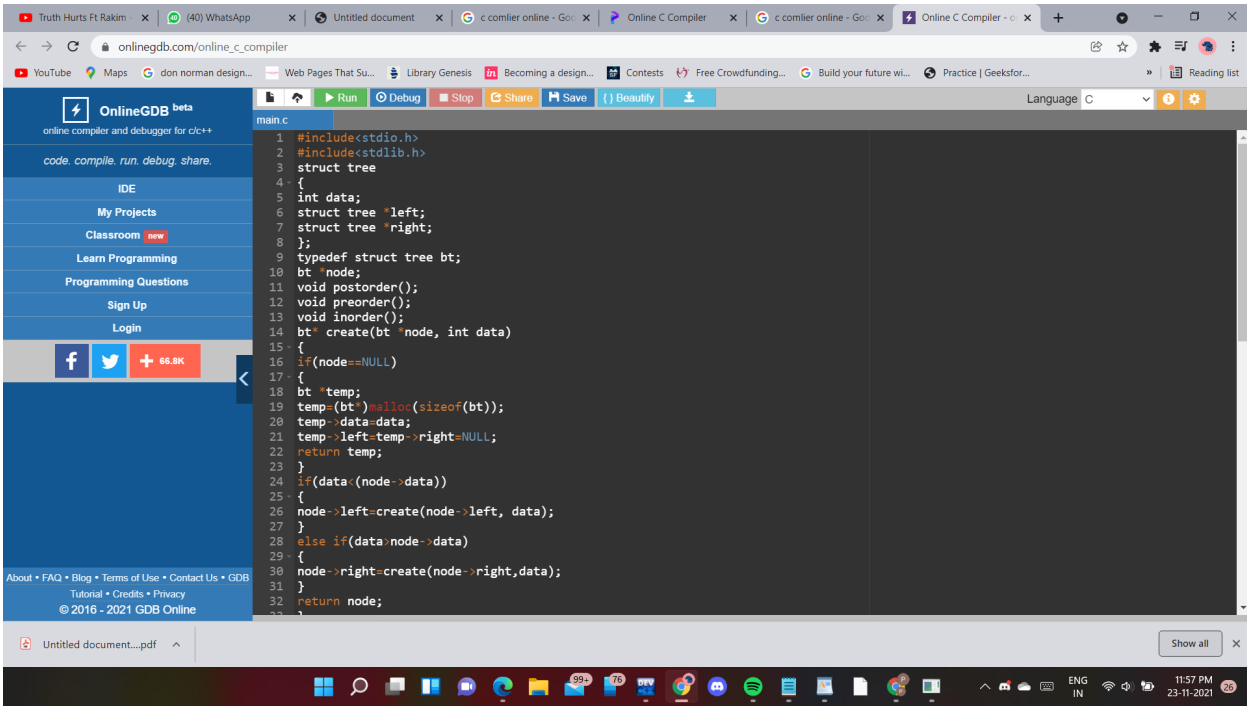
```
                enter data : 34

        press 1 if 34 has left node else 0  :0

        press 1 if 34 has right node else 0 :1

                enter data : 34

        press 1 if 34 has left node else 0  :1

                enter data : 67

        press 1 if 67 has left node else 0  :0

        press 1 if 67 has right node else 0 :0

        press 1 if 34 has right node else 0 :0

inorder traversal
root data : 34
root data : 67
root data : 34

...Program finished with exit code 0
Press ENTER to exit console.
```

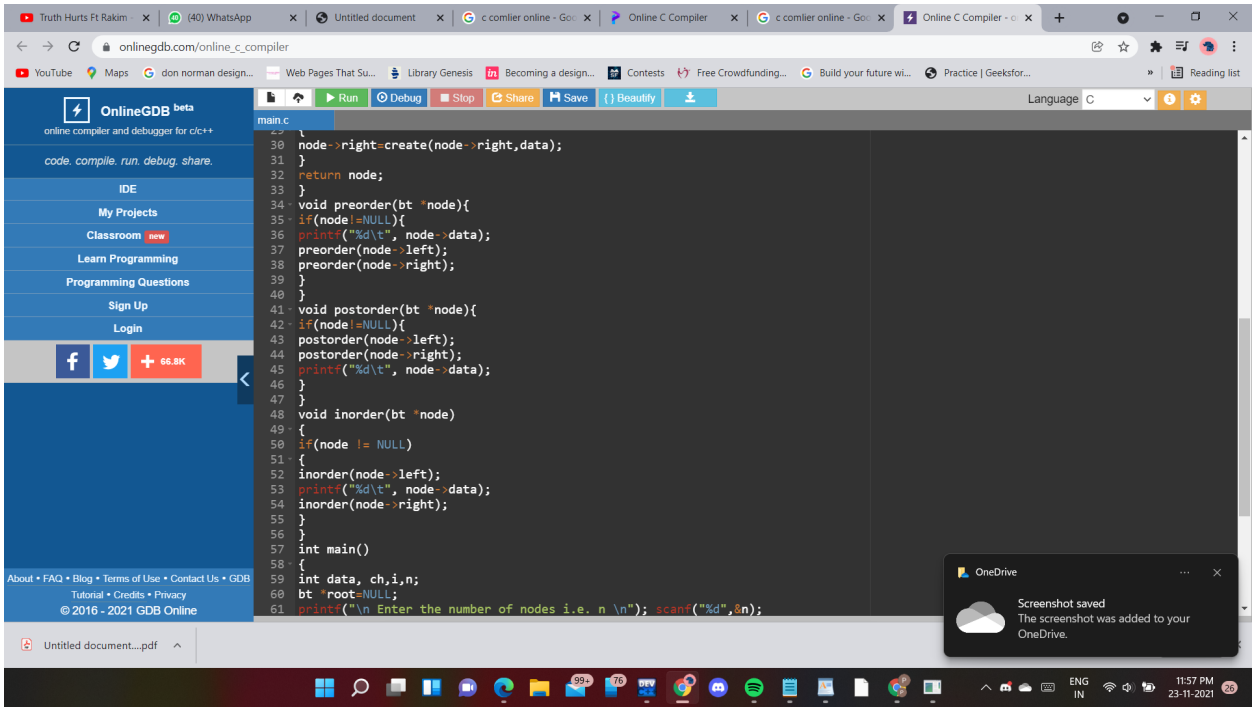
SOLUTION-2

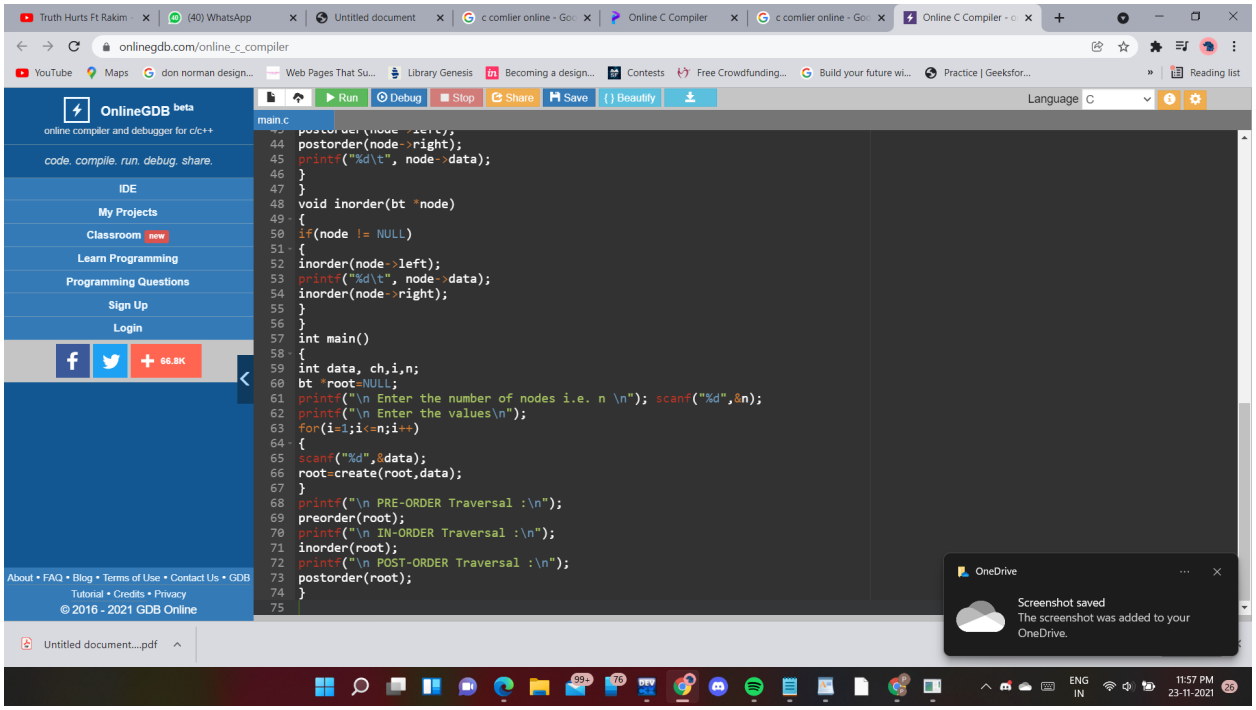
INPUT:

```
#include<stdio.h>
#include<stdlib.h>

struct tree
{
        int data;
        struct tree *left;
        struct tree *right;
};
typedef struct tree bt;

bt *node;

void postorder();
void preorder();
void inorder();
bt* create(bt *node, int data)
{
        if(node==NULL)
        {
                bt *temp;
                temp=(bt*)malloc(sizeof(bt));
                temp->data=data;
                temp->left=temp->right=NULL;
                return temp;
        }
        if(data<(node->data))
        {
                node->left=create(node->left, data);
        }
        else if(data>node->data)
        {
                node->right=create(node->right,data);
        }
        return node;
}

void preorder(bt *node){
   if(node!=NULL){
      printf("%d\t", node->data);
      preorder(node->left);
      preorder(node->right);
   }
}
```

```
void postorder(bt *node){
   if(node!=NULL){
      postorder(node->left);
      postorder(node->right);
      printf("%d\t", node->data);
   }
}


void inorder(bt *node)
{
        if(node != NULL)
        {
                inorder(node->left);
                printf("%d\t", node->data);
                inorder(node->right);
        }
}
int main()
{
        int data, ch,i,n;
        bt *root=NULL;
        printf("\n Enter the number of
        nodes i.e. n \n"); scanf("%d",&n);
        printf("\n Enter the values\n");
        for(i=1;i<=n;i++)
        {
                scanf("%d",&data);
                root=create(root,data);
        }
        printf("\n PRE-ORDER Traversal :\n");
        preorder(root);
        printf("\n IN-ORDER Traversal :\n");
        inorder(root);
        printf("\n POST-ORDER Traversal :\n");
        postorder(root);
}
```

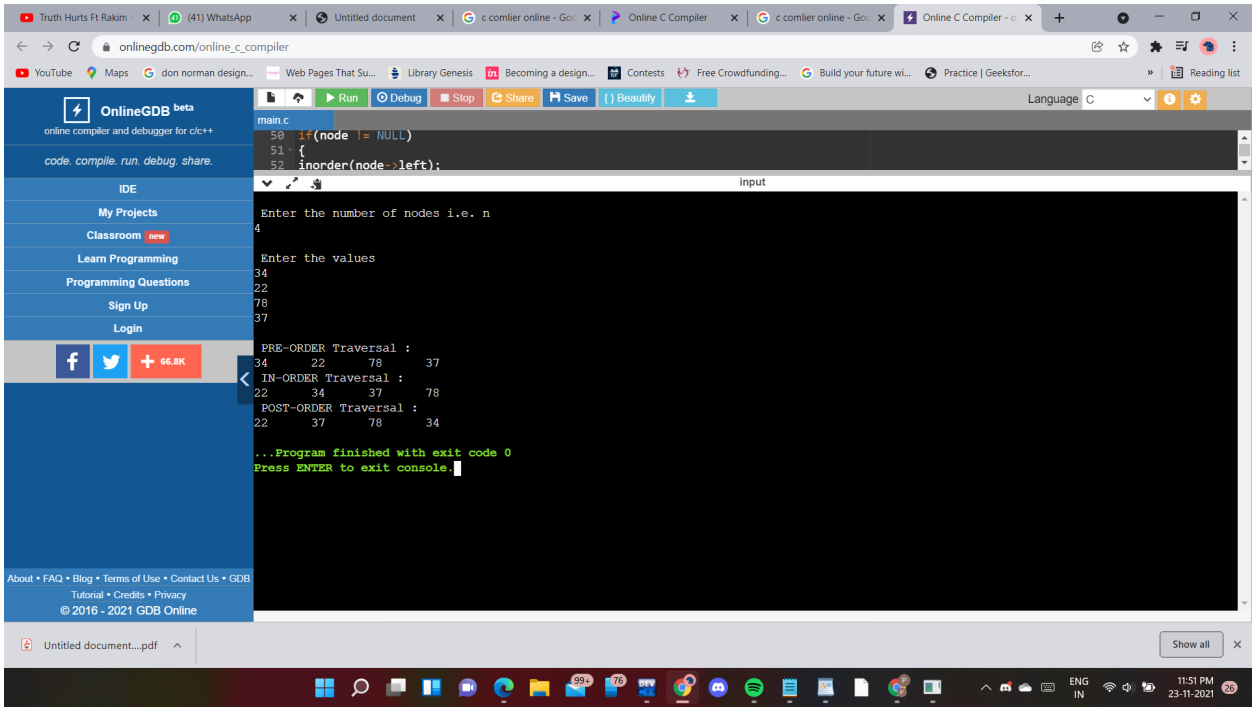

onlinegdb.com/online_c_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
66.8K
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
#include<stdio.h>
#include<stdlib.h>
struct tree
{
int data;
struct tree *left;
struct tree *right;
};
typedef struct tree bt;
bt *node;
void postorder();
void preorder();
void inorder();
bt* create(bt *node, int data)
{
if(node==NULL)
{
bt *temp;
temp=(bt*)malloc(sizeof(bt));
temp->data=data;
temp->left=temp->right=NULL;
return temp;
}
if(data<(node->data))
{
node->left=create(node->left, data);
}
else if(data>node->data)
{
node->right=create(node->right,data);
}
return node;
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Untitled document....pdf
Show all


onlinegdb.com/online_c_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
66.8K
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
node->right=create(node->right,data);
}
return node;
}
void preorder(bt *node){
if(node!=NULL){
printf("%d\t", node->data);
preorder(node->left);
preorder(node->right);
}
}
void postorder(bt *node){
if(node!=NULL){
postorder(node->left);
postorder(node->right);
printf("%d\t", node->data);
}
}
void inorder(bt *node)
{
if(node != NULL)
{
inorder(node->left);
printf("%d\t", node->data);
inorder(node->right);
}
}
int main()
{
int data, ch,i,n;
bt *root=NULL;
printf("\n Enter the number of nodes i.e. n \n"); scanf("%d",&n);
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Untitled document....pdf
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.



```
postorder(node->right);
printf("%d\t", node->data);
}
}
void inorder(bt *node)
{
if(node != NULL)
{
inorder(node->left);
printf("%d\t", node->data);
inorder(node->right);
}
}
int main()
{
int data, ch,i,n;
bt *root=NULL;
printf("\n Enter the number of nodes i.e. n \n"); scanf("%d",&n);
printf("\n Enter the values\n");
for(i=1;i<=n;i++)
{
scanf("%d",&data);
root=create(root,data);
}
printf("\n PRE-ORDER Traversal :\n");
preorder(root);
printf("\n IN-ORDER Traversal :\n");
inorder(root);
printf("\n POST-ORDER Traversal :\n");
postorder(root);
}
```

OUTPUT:



```
if(node != NULL)
{
inorder(node->left);

Enter the number of nodes i.e. n
4

Enter the values
34
22
78
37

PRE-ORDER Traversal :
34      22      78      37
IN-ORDER Traversal :
22      34      37      78
POST-ORDER Traversal :
22      37      78      34

...Program finished with exit code 0
Press ENTER to exit console.
```